A Complex Wavelet representation of Error Covariances in ALADIN 3d-Var

Alex Deckmyn

KMI, Belgium

Krakow, April 13, 2010

Contents

- Introduction
- A complex wavelet approach in 1+1 dimensions
- Full 3d implementation: first experiments
- Conclusions

Variational data assimilation

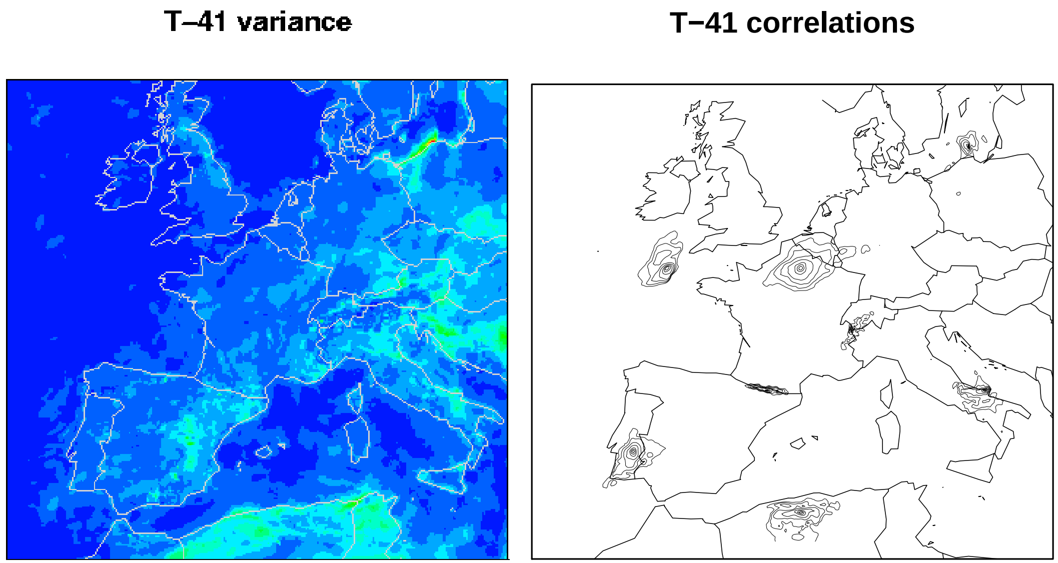
- In 3 dimensions:

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}^b)^T \mathbf{B}^{-1} (\mathbf{x} - \mathbf{x}^b) + \frac{1}{2}(\mathbf{H}\mathbf{x} - \mathbf{y})^T \mathbf{R}^{-1} (\mathbf{H}\mathbf{x} - \mathbf{y}),$$

- $\mathbf{H}$ is the observation operator. $\mathbf{B}$ and $\mathbf{R}$ are the covariance matrices of the background and observation errors, respectively.
- Formal exact solution:
$$\mathbf{x}^a = \mathbf{x}^b + \mathbf{B}\mathbf{H}^T \left(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R}\right)^{-1} (\mathbf{y} - \mathbf{H}\mathbf{x}^b)$$
- B describes how one observation influences the analysis in the neighbourhood.

Background Error Covariance

- The matrix **B** of background error covariances is vital to most assimilation methods.



- Heterogeneous, anisotropic structure functions (and often noisy).
- May be estimated in many ways, either statically or day by day (ensemble).

Simplifying B

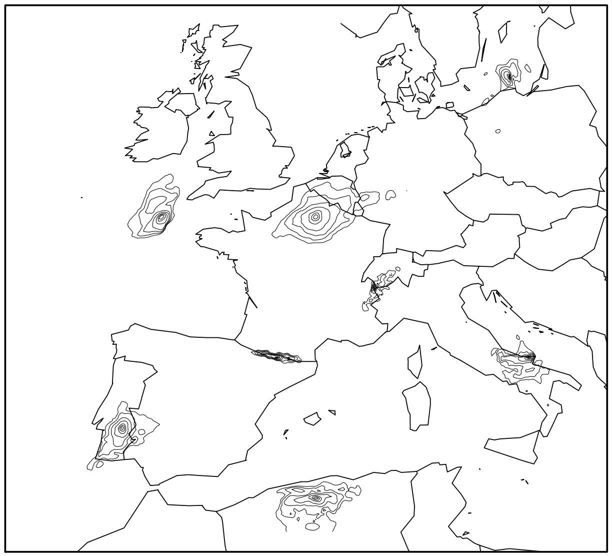
- **B** is much too large as a full matrix.
- A diagonal matrix requires much less memory (and inverting it is trivial).
- If we represent **B** in grid point space, the diagonal represents the error variance at every grid point. The correlation of errors in different points is completely ignored.

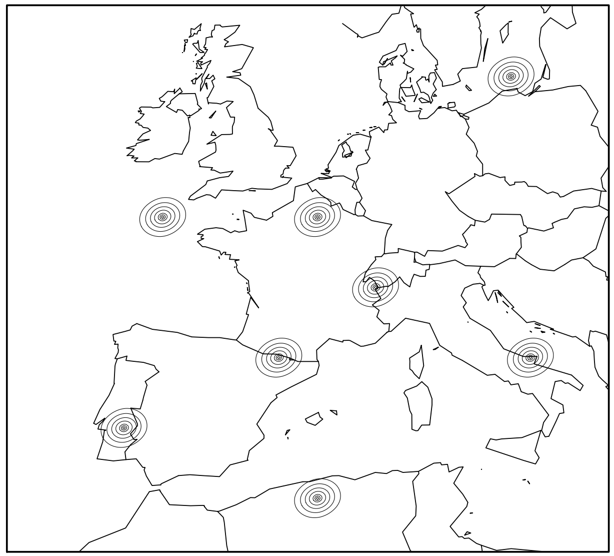
Diagonalising B

- We can represent our background field in spectral co-ordinates: $\mathbf{x}_f = \mathbf{F}\mathbf{x}$.
- The $\mathbf{B}$ matrix in Fourier space is $\mathbf{B}_f = \mathbf{F}\mathbf{B}\mathbf{F}^*$.
- If we diagnolise $\mathbf{B}_f$, the covariance matrix in grid point space becomes $\tilde{\mathbf{B}} = \mathbf{F}^*\mathbf{B}_f\mathbf{F}$
- A diagonal matrix in Fourier space corresponds to homogeneous structure functions in grid space.
- One may combine these 2 representations $\mathbf{B} = \mathbf{D}_g\mathbf{F}^*\mathbf{C}_f\mathbf{F}\mathbf{D}_g$, where $\mathbf{D}_g$ represents the standard deviation and $\mathbf{C}_f$ the correlations.

Spectral diagonalisation

- Spectral diagonalisation of $\mathbf{B}$ gives homogeneous correlation functions:





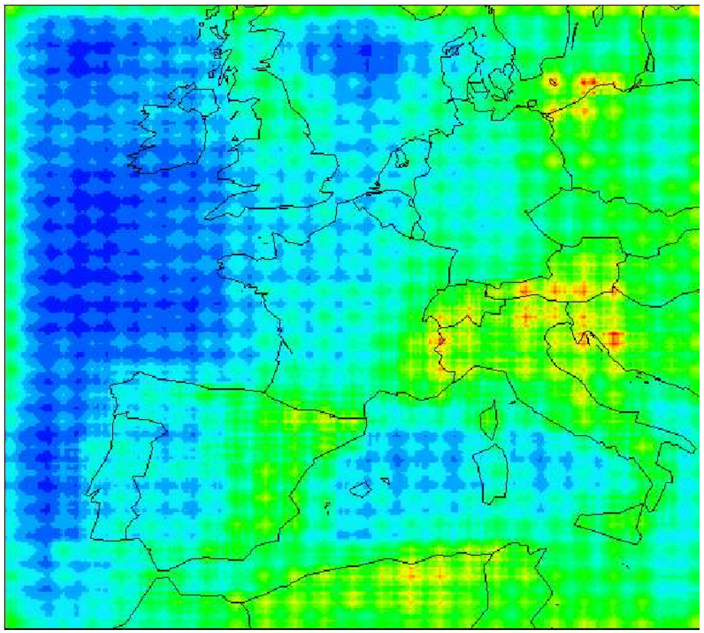
- One "average" structure function for the whole domain.
- Can we do better than this?

Introducing wavelets

- M. Fisher (2001) introduced the idea of representing $\mathbf{B}$ in wavelet co-ordinates.
- For the *global* model IFS used at ECMWF he introduced a set of non-orthogonal, band-limited wavelets.
- No longer an orthogonal basis, but a tight frame.

A first (naive) approach

- Using 1 single orthonormal basis of (Meyer) wavelets, the variance (diagonal of $\mathbf{B}$) in grid point space becomes:



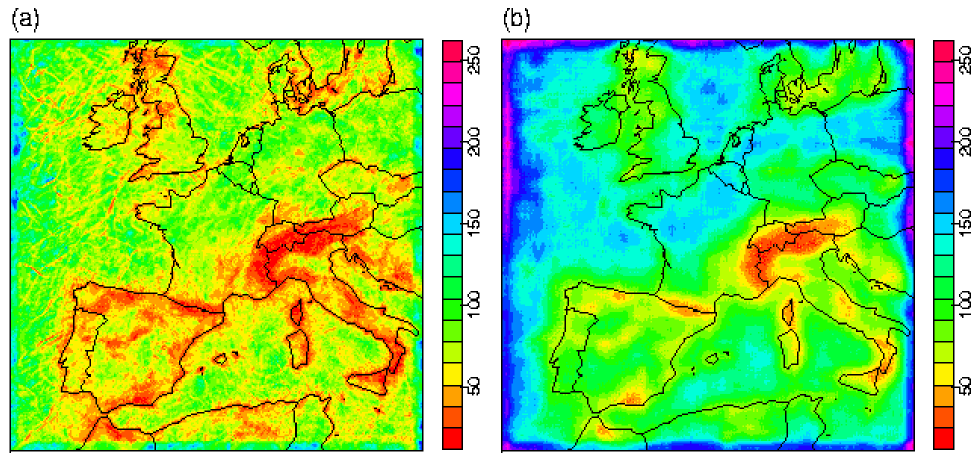
- The wavelet transform & diagonalisation introduce inacceptable artifacts.

A hybrid Meyer approach

- In Deckmyn & Berre (2005) wavelets were combined with grid point and Fourier.
- Use every basis for its strongest points:
  - Grid space: strictly local (variance)
  - Fourier space: average correlation function
  - Wavelet space: local differences from average
- $\mathbf{B} = \mathbf{D}_g^* \mathbf{F}^* \mathbf{D}_f^* (\mathbf{F}^{-1})^* \mathbf{W}^* \mathbf{B}_w \mathbf{W} \mathbf{F}^{-1} \mathbf{D}_f \mathbf{F} \mathbf{D}_g,$

A hybrid Meyer approach

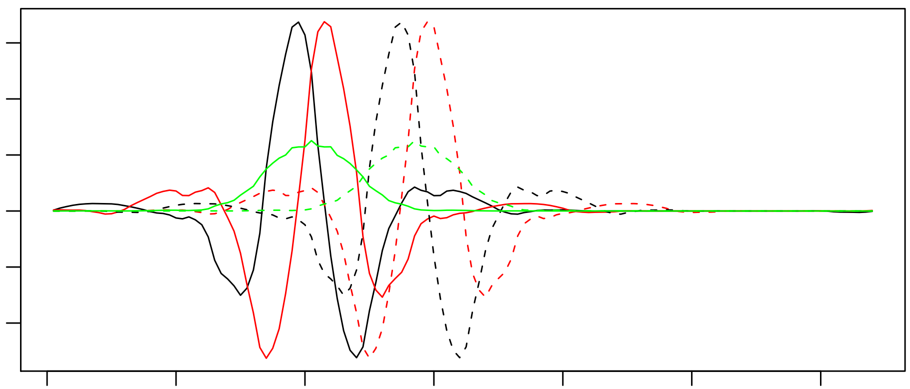
- The length scale (whidth of the local correlation function $L^2 \approx \frac{-2\rho}{\nabla^2\rho}$) becomes:



- In fact a clearer image than the noisy original!
- Quite cumbersome. A lot of calculations for relatively small gain.

Complex wavelets

- Introduced by N. Kingsbury (2001).
- 2 separate ("Q-shift dual tree") orthonormal wavelet transforms, that can be interpreted as real and imaginary components ($\approx$ windowed cos & sin)



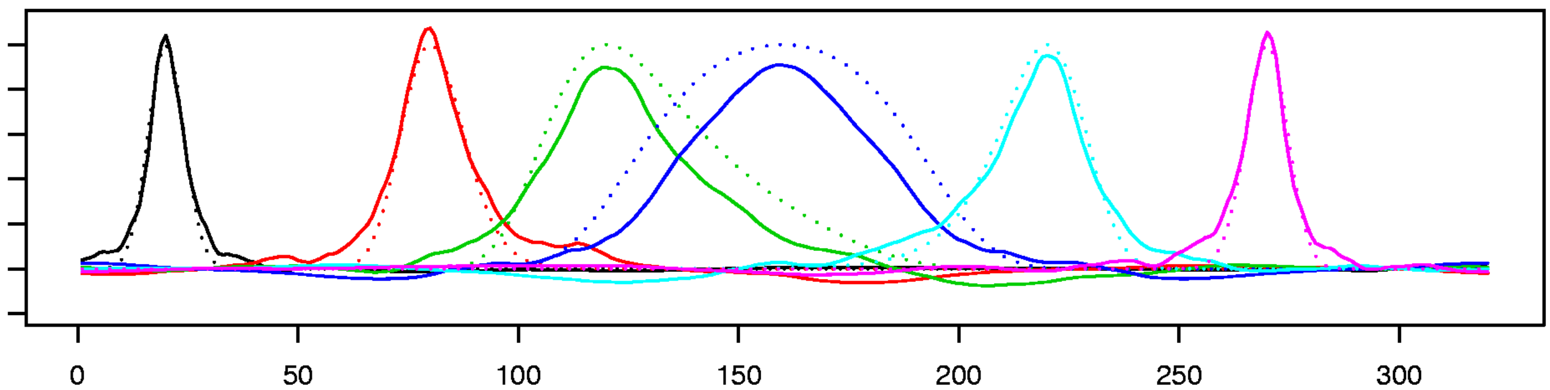
- Approximate shift invariant $\rightarrow$ much less artifacts!
- Limited redundancy of 2:1.

Complex wavelets

- The two wavelet filters are about 1/2 sample apart.
- To achieve this, the first stage of $\mathbf{W}_2$ in fact uses a different filter.
- The complex wavelet transform $\mathbf{W} = \mathbf{W}_1 + i\mathbf{W}_2$ (or, equivalently, the combination of $\mathbf{W}_1$ and $\mathbf{W}_2$ as a set of 2N real functions) is not orthonormal, but a tight frame of multiplicity 2.
- The two wavelets are in fact the reverse of each other!
- $\mathbf{B}_w$ is a complex, hermitian $N \times N$ matrix.

1D Complex wavelets

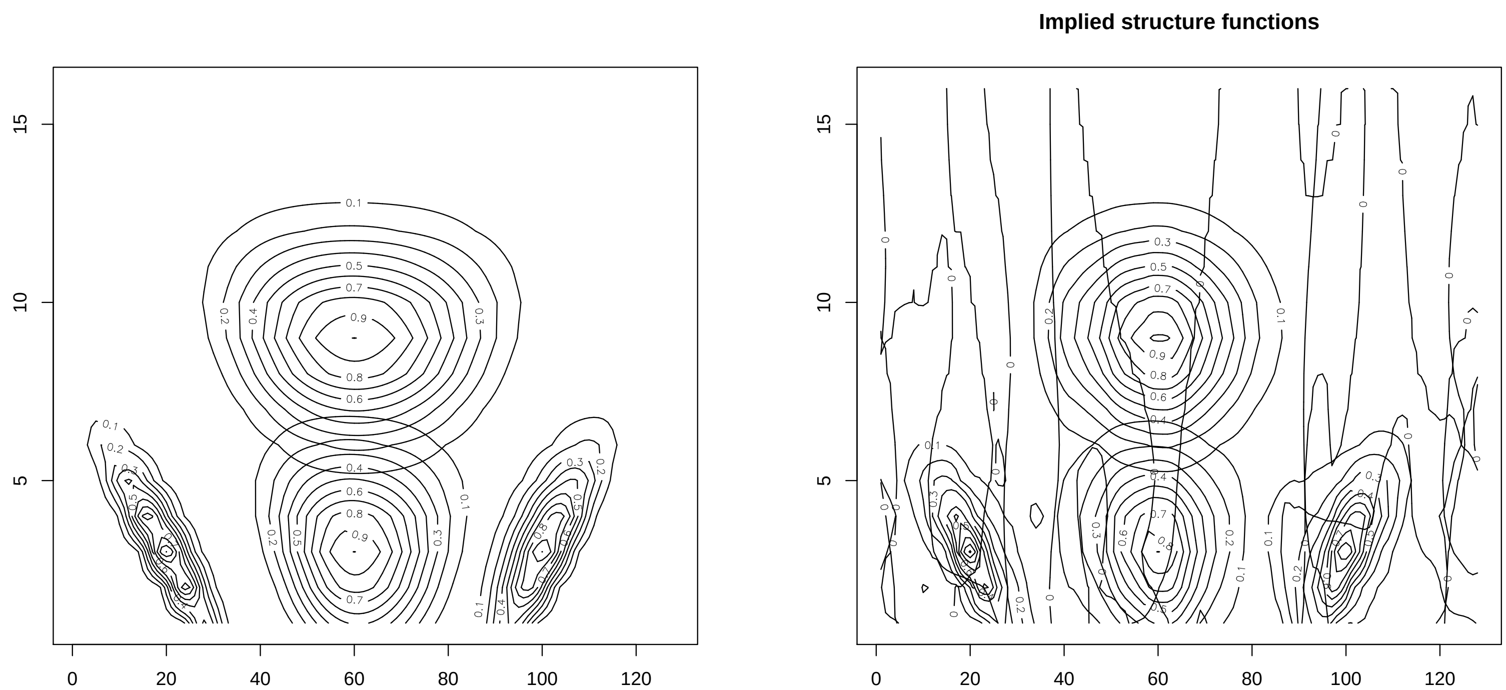
- A synthetic 1D example: 1D structure functions, original (dotted lines) and modelled with complex wavelets (bold).



- Variance tends to be underestimated in regions with large length scales. This has also been observed by other researchers.

1D Complex wavelets

- Because the covariances are complex (the real part at level 1 can be correllated to the imaginary part at level 2), the diagonal **B** can still represent tilted structures, with some limitations.



Non-periodic boundaries?

- A single wavelet transform allowing symmetric boundaries, must be symmetric and odd length.
- (Anti-)symmetric wavelets can not be orthogonal (except the Haar wavelet).
- For the complex wavelets, symmetric wavelets are made possible by the combination of two inverse wavelets. The “errors” at the periodic boundaries thus compensate. BUT the first (smallest) scale needs a different wavelet.
- Kingsbury uses a symmetric, non-orthogonal wavelet. Others propose an orthogonal, even length, almost-symmetric filter, but this only allows periodic boundaries.

Non-periodic: solution

- Consider not the original domain $X[1..N]$, but the doubling $X[1..N, N..1]$, which is periodic by definition.
- On this domain, find an orthogonal almost-symmetric & odd length wavelet for the first stage. Such a wavelet existed in literature, but was hard to find.
- Use Kingsbury's wavelet on the next stages.
- Elements 1..N are the real part, the second half are the corresponding imaginary parts.
- So the resulting solution is no longer a combination of 2 orthogonal wavelets, but 1 transform on the doubled domain!

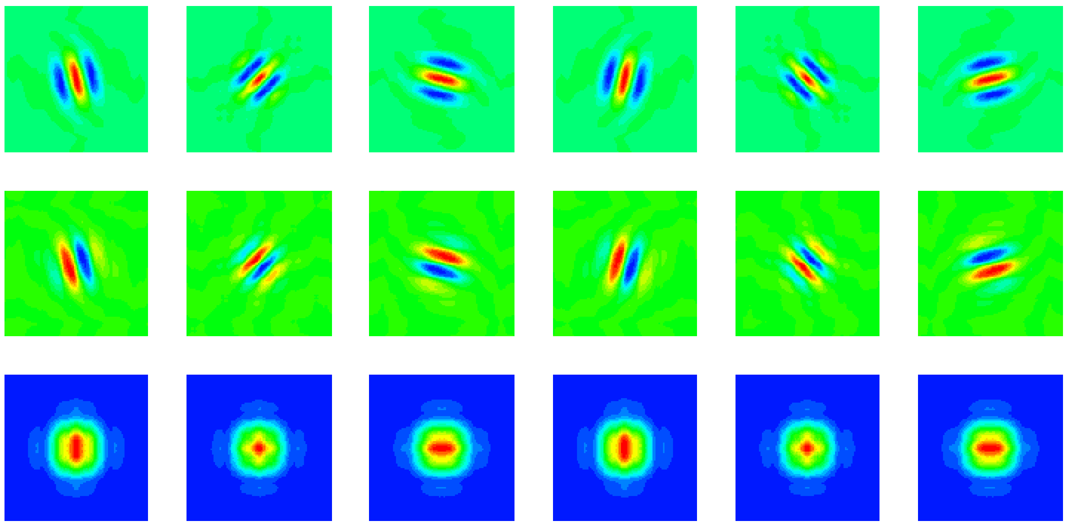
2D complex wavelets

- Tensor products of the 2 dual-tree wavelets yields 4 different orthonormal 2D wavelet transforms, each of which has 3 different sectors.
- By taking certain linear combinations of these wavelet functions, we get a new (non-orthogonal) set of complex wavelets:

$$
\begin{aligned}
\Psi_l^1 &= (\psi_l^{11} + \psi_l^{22}) + i(\psi_l^{12} - \psi_l^{21}), \\
\Psi_l^2 &= (\psi_l^{12} + \psi_l^{21}) + i(\psi_l^{11} - \psi_l^{22}).
\end{aligned}
$$

2D complex wavelets – orientations

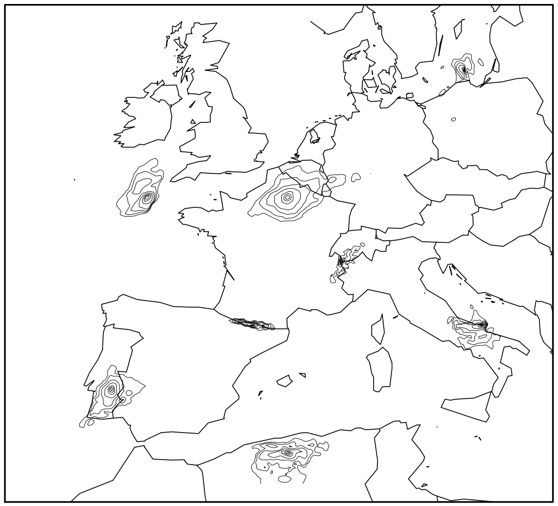
- Real part, imaginary part and modulus of the 2D wavelets:

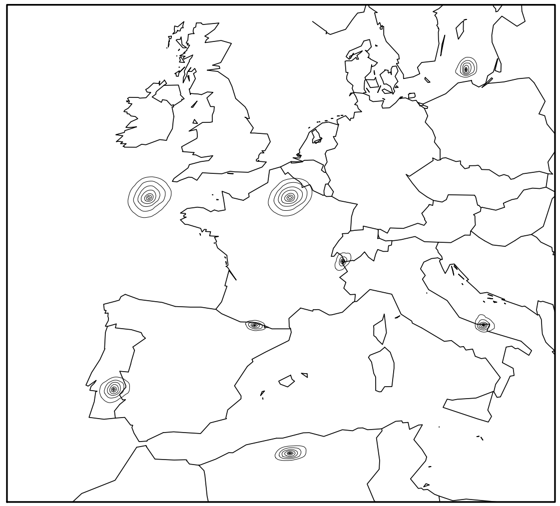


- →6 distinct directional components at every scale!

Complex wavelet transform

- If we use these directional wavelets to diagonalise **B**:





3-dimensional covariances

- We now move on to an $N \times N \times L$ grid.
- Vertical covariances are best treated in a different manner: e.g. they are not evenly spaced. Also, L is usually much smaller than N (e.g. 46).
- In the standard approach, $\mathbf{B}$ becomes block-diagonal: for every 2D component (spectral or wavelet) we get a "vertical" $L \times L$ block.
- In the assimilation code, the eigenvectors of these blocks are computed numerically.

Reducing memory cost

- For 1 field, we still have $4N^2 \times L^2$ components, which is too much in an operational setting.
- In ALADIN, the spectral components are also averaged by (total) wave number. So in 2D, there are only $O(N)$ components left.
- We introduced 2 reductions.
- In the horizontal components, we may eliminate the smallest scales.

Reducing the vertical matrices

- For every scale and orientation (denoted l) we calculate the *mean vertical matrix* $\overline{\mathbf{B}}_l$.
- This matrix can be rewritten in its eigenbasis $\overline{E}_l$:

$$\overline{\mathbf{B}}_l = \overline{E}_l \overline{\Lambda}_l \overline{E}_l^*$$

- We then assume that the basis of eigenvectors $\overline{E}_l$ is representative for all local eigenbases. Thus we write for location i:

$$\mathbf{B}_{l,i} \approx \overline{E}_l \Lambda_{l,i} \overline{E}_l^*,$$

where $\Lambda_{l,i} = diag(\overline{E}_l^* \mathbf{B}_{l,i} \overline{E}_l)$.

Reducing the vertical matrices

- This averageing of vertical components has a rather strong smoothing effect. Some detail (especially close to the surface) is lost.
- This can be (partially) solved by taking out the variance before the reduction, thus only reducing the correlations, which are usually smoother.

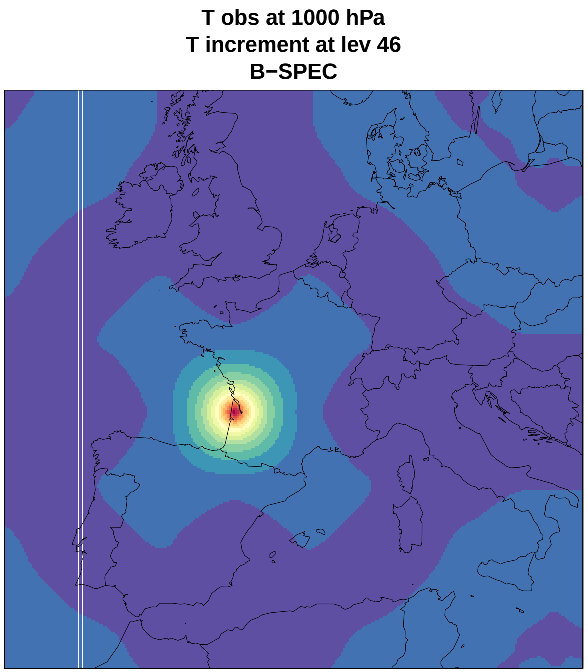
Implementation in ALADIN

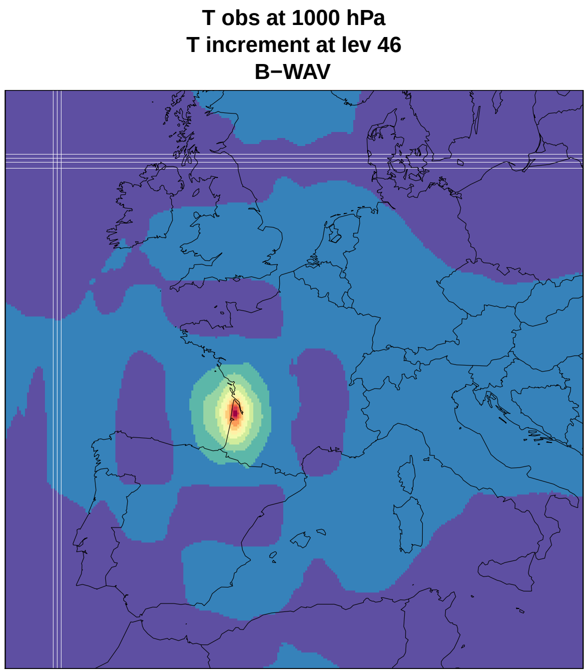
- Full wavelet treatment of unbalanced part of B. The statistical balance is still in Fourier space.
- Currently, only periodic wavelet version coded.
- Can be turned on with the switch LJBWAVELET.
- Still very experimental.
- B_wav is calculated offline using a wavelet version of FESTAT.

Implementation in ALADIN

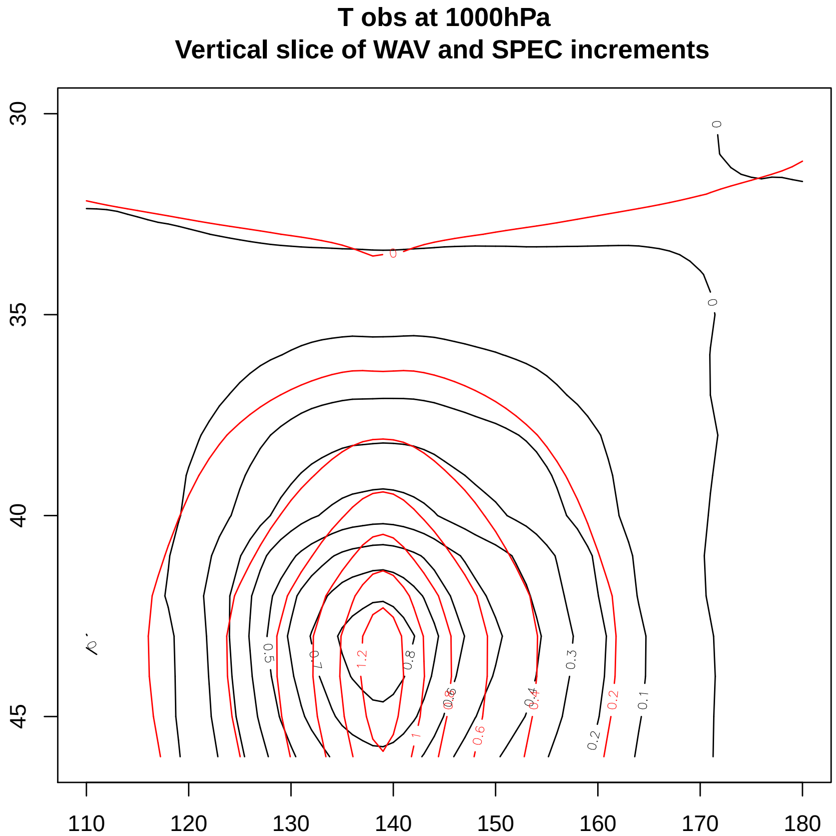
- Wavelet domain must contain enough powers of 2 (so extend $300 \longrightarrow 320$)
- For the balance part, we go back to Fourier space.
- This requires bi-periodicisation at every iteration.
- Not only expensive, also no simple adjoint.
- Still causes noise unless we use the same (larger) extension zone for Fourier space as well.

Single T obs at 1000hPa



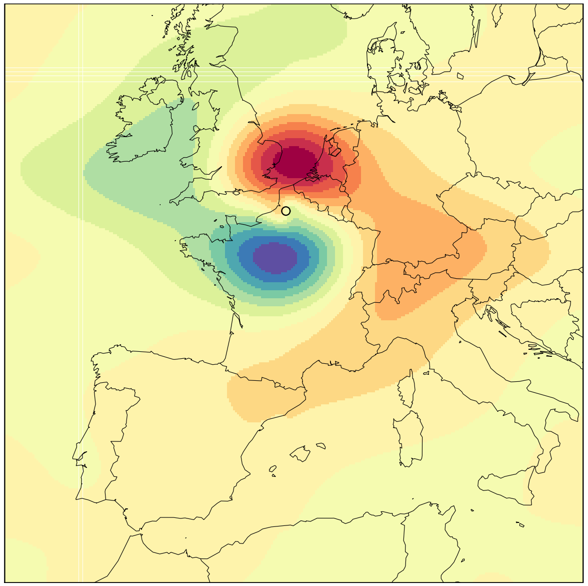


Single T obs at 1000hPa

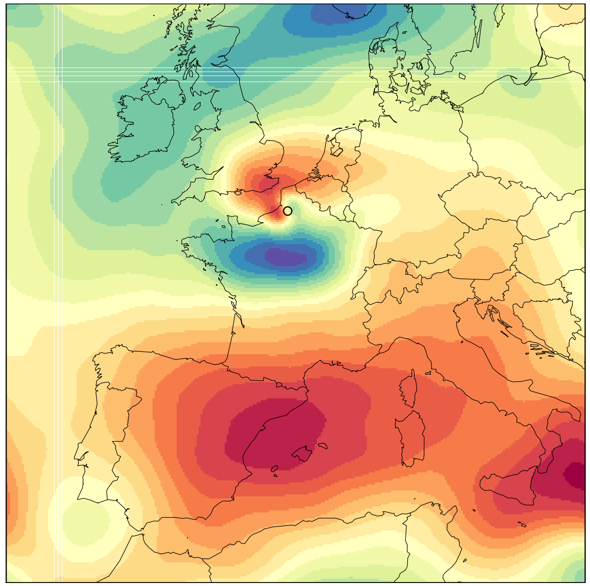


Problem with wind increment?

T obs at 300 hPa
U increment lev 20
B−SPEC



T obs at 300 hPa
U increment lev 20
B−WAV



Conclusions: advantages

- Using wavelet transform we can simplify the $\mathbf{B}$ matrix while maintaining its basic local and spatial (scale) features.
- Compared to standard ODWT, dual-tree complex wavelets have several advantages:
  - Approximate shift invariance (and hence reduced “artifacts”).
  - Improved directional resolution,
  - possibility of symmetric boundaries,
  - A phase that allows for tilted structure functions in 3D... But maybe it is removed by the vertical approximations...

To do

- Find out what is happening with the wind increments.
- Implement non-periodic transforms in ALADIN.
- Much more experimenting.
- Solve remaining noise issues & normalisation of variances.
- Better vertical representation (tilt,...)?